



Lab - Fault Handling

Wherein ...

- We are no longer satisfied with allowing any and all requests to hit our back end server, and
- Want to specify precisely which requests may be sent to our server, and
- Want to change the Error Code and Message delivered in certain circumstances

Create a FaultRule

- Create a fault rule in Proxy Endpoint
- Make sure the fault.name variable exactly matches the 'QuotaViolation' string

```
<FaultRules>  
  <FaultRule name="QuotaViolation">  
    <Condition>fault.name == "QuotaViolation"</Condition>  
    <Step><Name>QuotaViolation</Name></Step>  
  </FaultRule>  
</FaultRules>
```

Implement the QuotaViolation policy

- Create a Assign Message Policy named “Quota-Violation”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <AssignMessage async="false" continueOnError="false" enabled="true"
    name="QuotaViolation">
    <DisplayName>QuotaViolation</DisplayName>
    <FaultRules/>
    <Properties/>
    <Set>
      <Headers/>
      <Payload contentType="application/json">
        {"error": "Quota Violation. Too many requests."}
      </Payload>
      <StatusCode>429</StatusCode>
      <ReasonPhrase>Violation</ReasonPhrase>
    </Set>
    <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
    <AssignTo createNew="false" transport="http" type="request"/>
  </AssignMessage>
```

Raising a Fault

- Add a "default" flow to block all un-supported resource requests
- Raise a fault from this flow

```
<Flows>
  <Flow name="default">
    <Description/>
    <Request>
      <Step>
        <Name>Raise-Fault-1</Name>
      </Step>
    </Request>
    <Response/>
  </Flow>
</Flows>
```

Implement the RaiseFault policy

- Add a new fault rule to handle the unknown resource paths:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RaiseFault async="false" continueOnError="false" enabled="true" name="Raise-Fault-1">
  <DisplayName>Raise Fault-1</DisplayName>
  <Properties/>
  <FaultResponse>
    <Set>
      <Headers/>
      <Payload contentType="text/plain">
        Method "..{proxy.pathsuffix}" Not Found
      </Payload>
      <StatusCode>404</StatusCode>
      <ReasonPhrase>Not Found</ReasonPhrase>
    </Set>
  </FaultResponse>
  <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
</RaiseFault>
```

Add conditional flows

- Add conditional flows for the pathsuffixes we know about (to be used later)

```
<Flow name="chefs">  
  <Description/>  
  <Request/>  
  <Response/>  
  <Condition>(proxy.pathsuffix MatchesPath "/chefs**") and (request.verb = "GET")</Condition>  
</Flow>
```

```
<Flow name="trucks">  
  <Description/>  
  <Request/>  
  <Response/>  
  <Condition>(proxy.pathsuffix MatchesPath "/trucks**") and (request.verb = "GET")</Condition>  
</Flow>
```

THANK YOU